US 20060062458A1

(54) **METHOD AND SYSTEM FOR IDENTIFYING OBJECTS IN AN IMAGE**

(75) Inventor: **Harry C. Lee**, Maitland, FL (US)

Correspondence Address:
**BUCHANAN INGERSOLL PC
(INCLUDING BURNS, DOANE, SWECKER & MATHIS)
POST OFFICE BOX 1404
ALEXANDRIA, VA 22313-1404 (US)**

(73) Assignee: **Lockheed Martin Corporation**

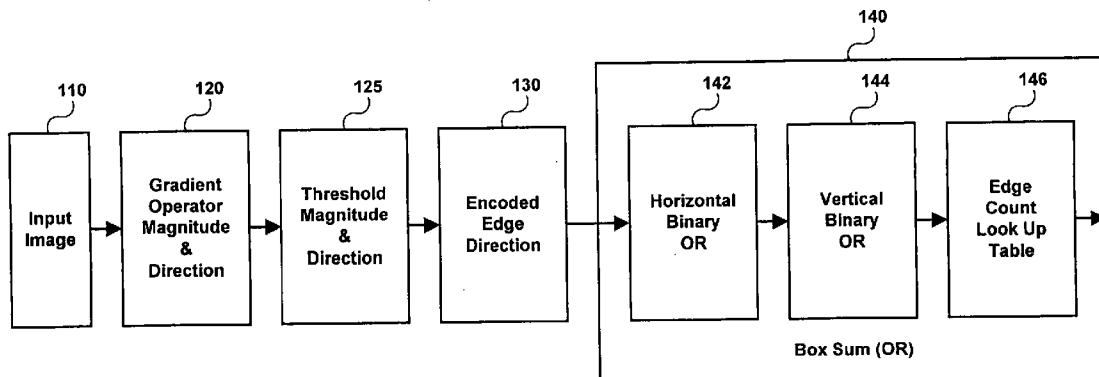**Publication Classification**

(57) **ABSTRACT**

Methods and apparatus for identifying objects in an image include processing the image with a gradient operator to produce a gradient magnitude and direction for each pixel. A number of different gradient directions in a portion of the processed image is determined. The portion of the processed image is identified as an object if the number of different gradient directions exceeds a threshold number of gradient directions. The determination of the gradient directions can be simplified by employing a lookup table.

Figure 1

a	b	c
d	e	f
g	h	i

Figure 3B

Figure 3A

Input Image

Sobel Edge Directions Image

220

Sobel Edge Magnitude Image

230
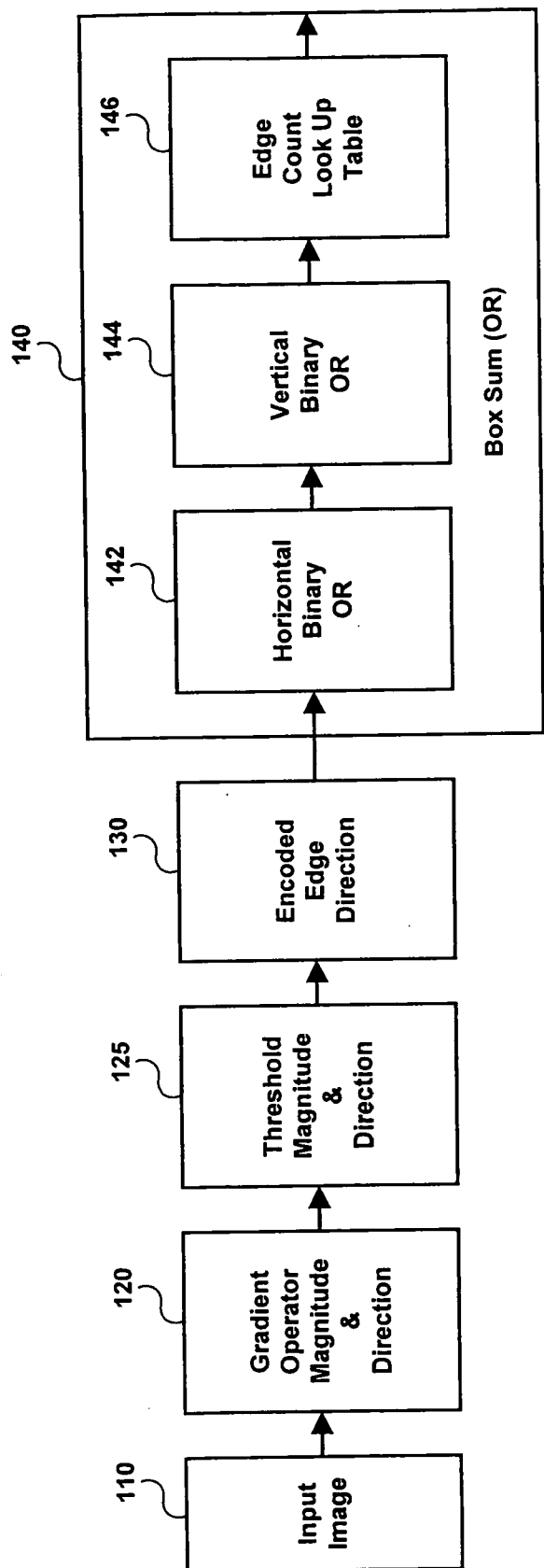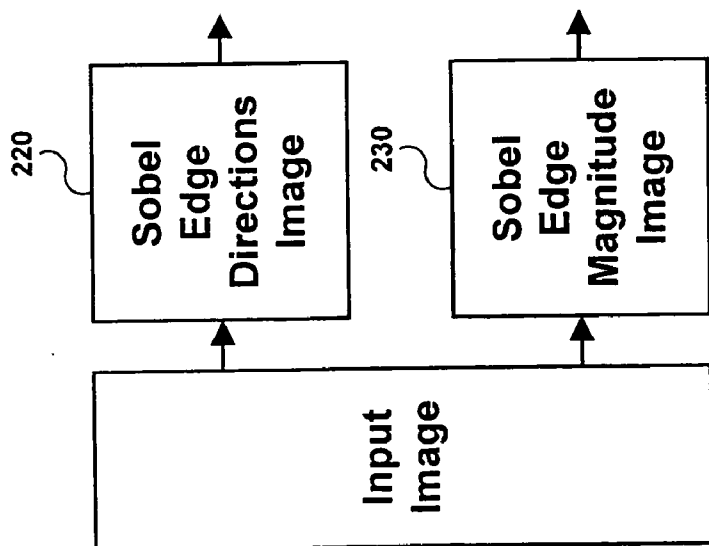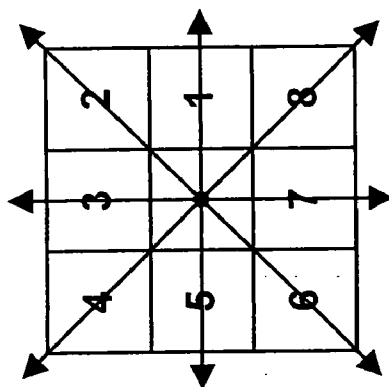
Figure 2

Edge Direction	Bit Encoded Edge Direction	Value of Encoded Edge Direction
1	00000001	1
2	00000010	2
3	00000100	4
4	00001000	8
5	00010000	16
6	00100000	32
7	01000000	64
8	10000000	128

Figure 4

540

Output
Binary
Box OR

530

Binary OR

Horizontal

520

Binary OR

Vertical

510

Encoded
Gradient
Directions

Image
Portion

Figure 5

Figure 6

Bit Encoded Direction

**Number of Different Directions
Look Up Table (LUT)**

Index	Output
. . .	. . . .
132	2
133	3
134	3
135	4
136	2
137	3
. .	. .

3 → | 00000100 |

7 → | 01000000 |

OR

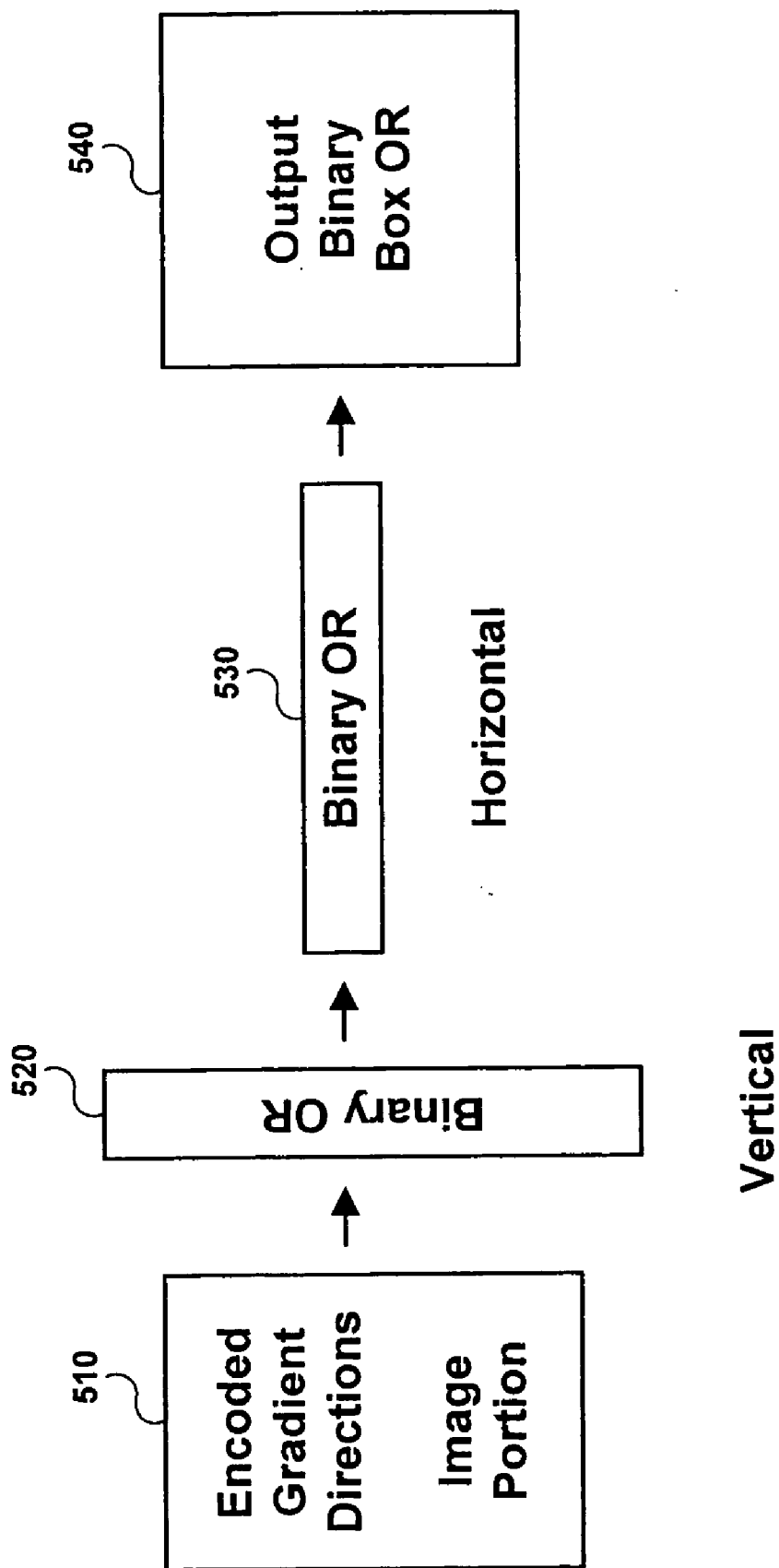| 01000100 |

136

**Or of Bit
Encoded
Direction**
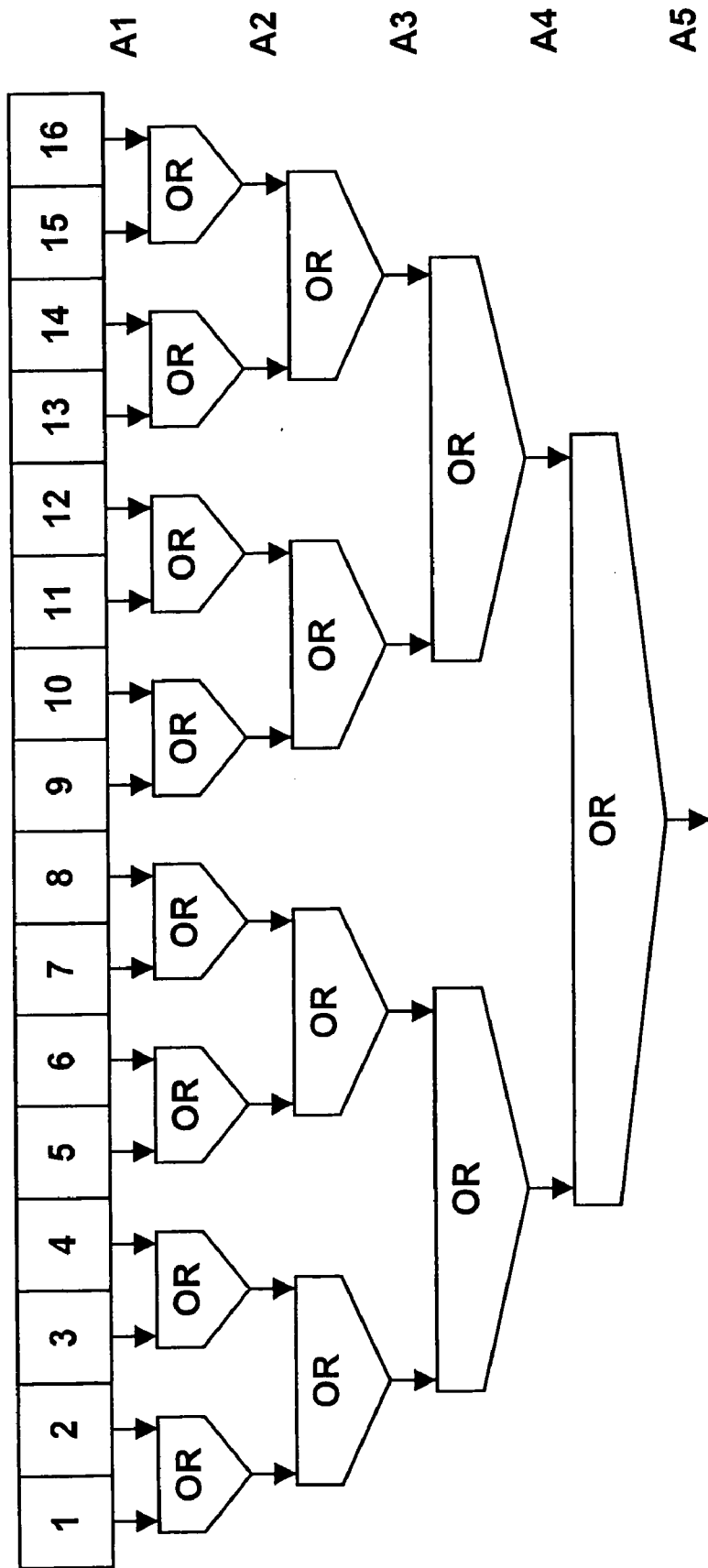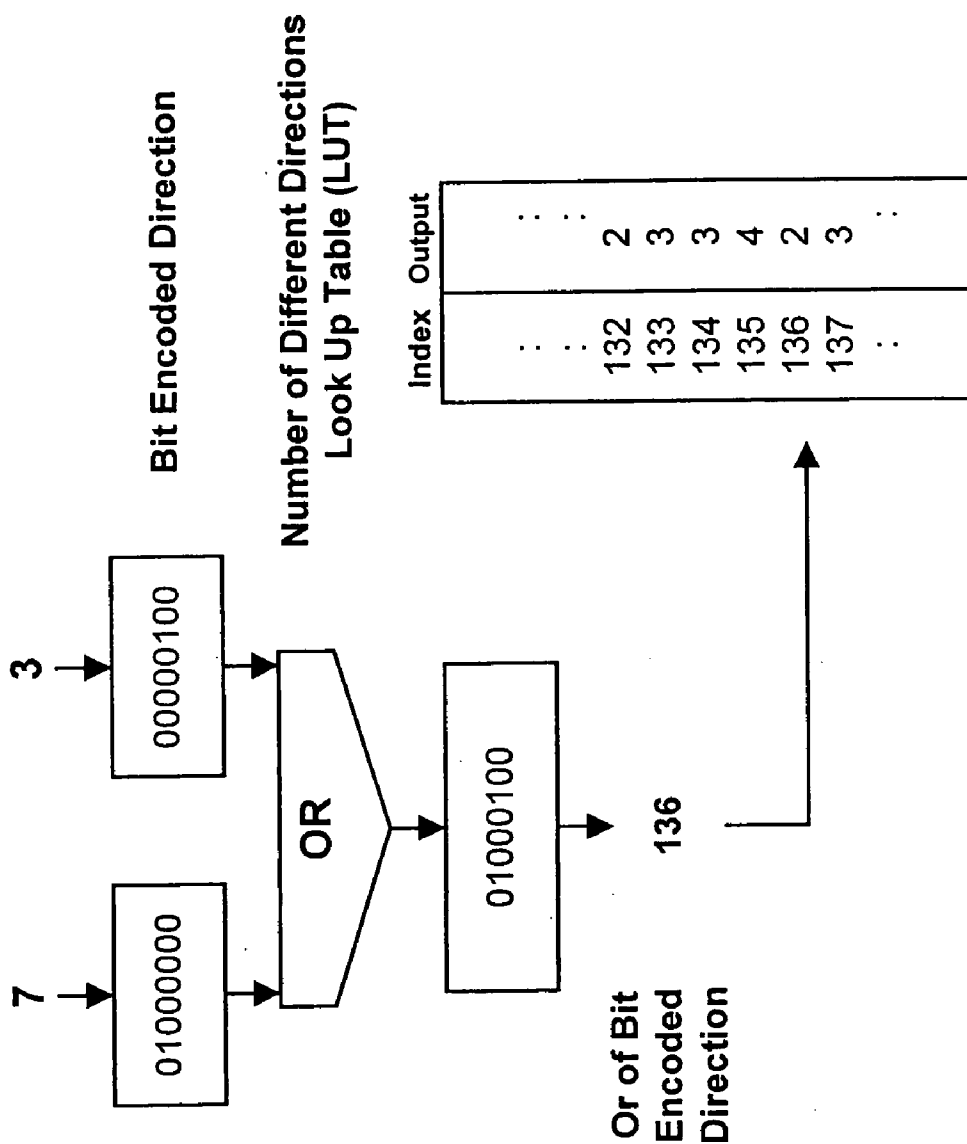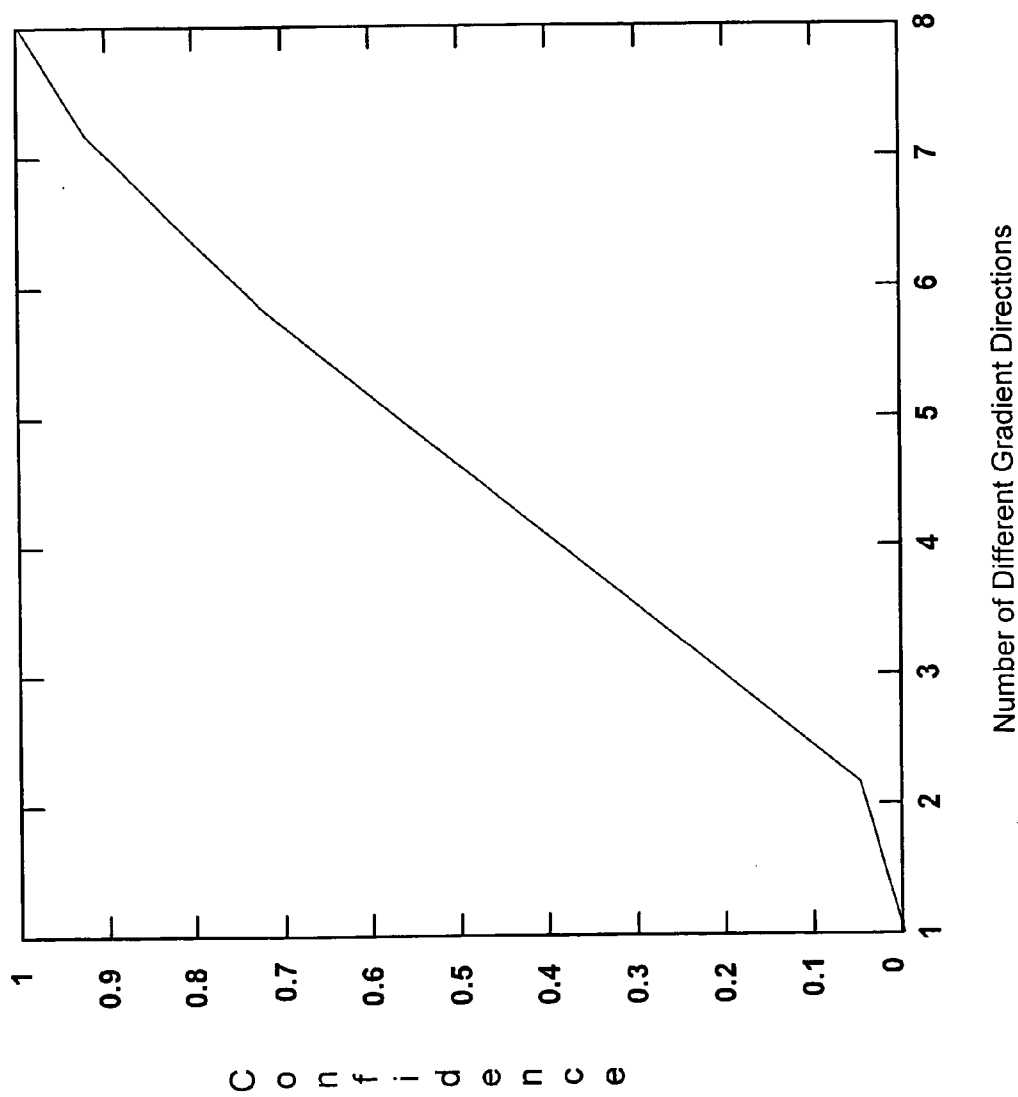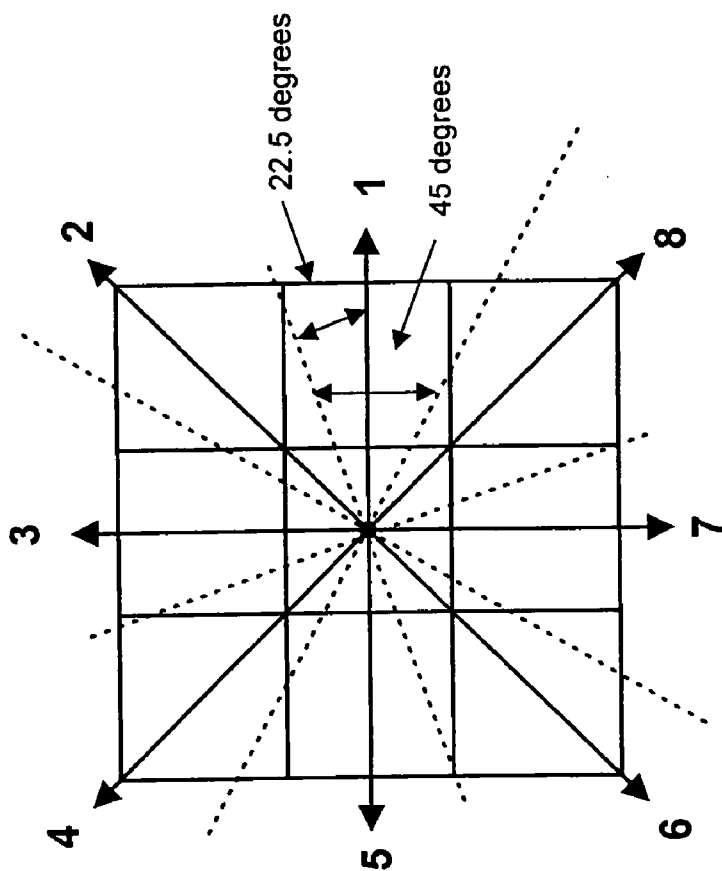
Figure 7

Figure 8

Figure 9B
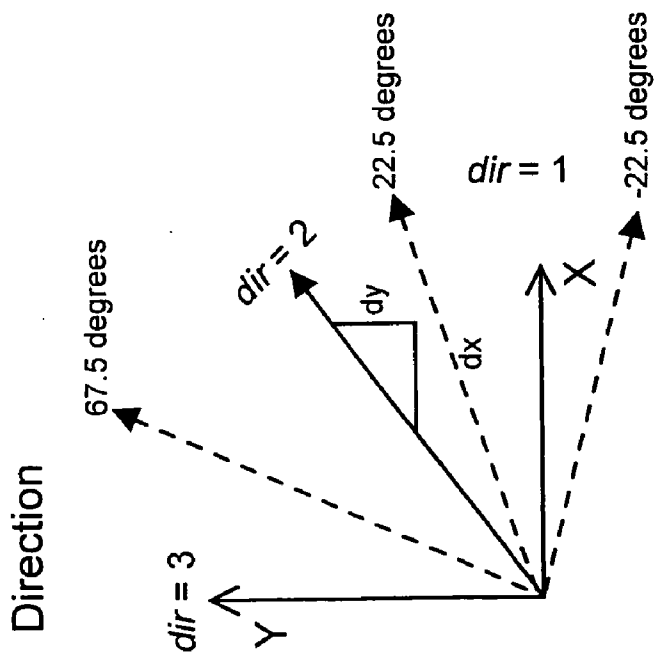
Figure 9A
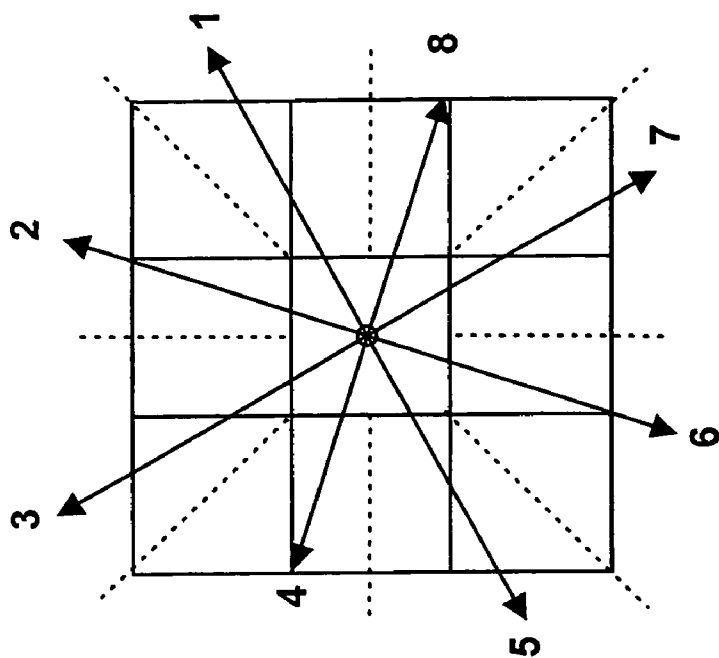
Figure 10B
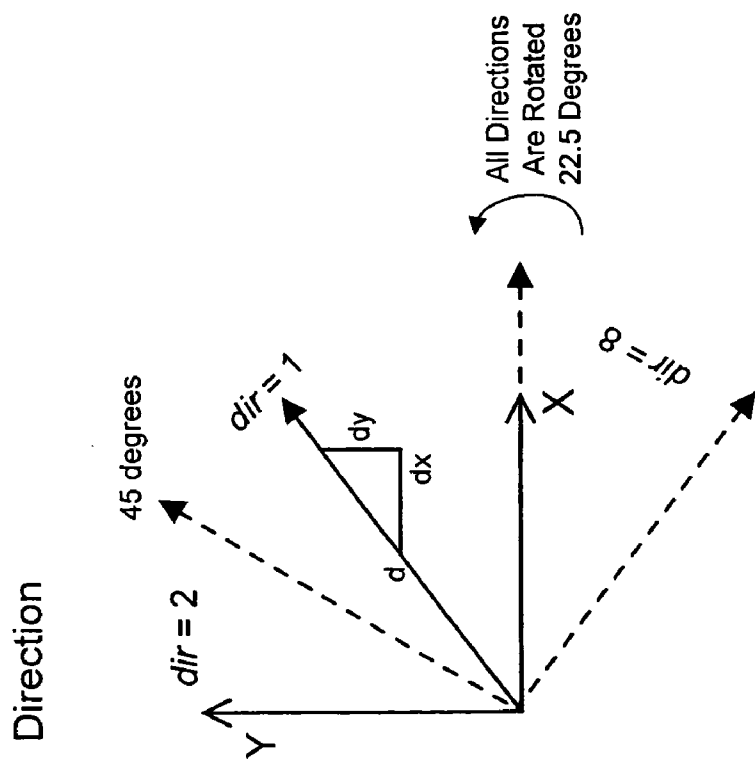
Figure 10A

Dy>=0	Dx>=0	Abs(Dx) > Abs(Dy)	OR'ed Value	Direction
000	000	000	000	6
000	000	001	001	5
000	010	000	010	7
000	010	001	011	8
100	000	000	100	3
100	000	001	101	4
100	010	000	110	2
100	010	001	111	1

Figure 11

METHOD AND SYSTEM FOR IDENTIFYING OBJECTS IN AN IMAGE

BACKGROUND

[0001] 1. Field of Invention

[0002] The present invention relates to processing of image data. More particularly, the present invention relates to a method and apparatus for identifying objects in an image.

[0003] 2. Background Information

[0004] Historically, reconnaissance information has provided important information used in planning military operations. For example, prior to the advent of photography, scouts would be sent out to collect information regarding natural resources such as lakes and rivers, enemy troop information and the like. With the advent of photography, these scouts would provide reconnaissance information by capturing a scene of enemy installations, battlefields, and the like, using photographs. As technology advances, new methods are provided for collecting reconnaissance information. For example, it is quite common today to have reconnaissance planes, manned or remotely controlled, or satellites capture a scene for reconnaissance purposes. In addition to conventional photographic techniques, a scene can be captured using infrared detectors and the like.

[0005] Typically scenes captured by reconnaissance techniques have been analyzed by humans in order to determine the content of the captured scene. For example, a human would analyze a photograph to determine the location of bodies of water, the location of enemy troops and the location of man-made objects such as buildings and lines of communication. The human who analyzed the photograph would then have to relay the determined information to people in the field, for example, to an airplane pilot in order to identify targets. However, using humans to analyze photographs is very labor intensive. Further, there can be a considerable delay between the time when a scene is captured and the time in which the information in the captured scene is relayed to persons in the field.

SUMMARY OF THE INVENTION

[0006] In accordance with a first exemplary aspect of the present invention a method and apparatus for identifying objects in an image is provided. In accordance with the aspect the image is processed with a gradient operator to produce a gradient magnitude and direction for each pixel. A number of different gradient directions in a portion of the processed image are determined. The portion of the processed image is identified as an object if the number of different gradient directions exceeds a threshold number of gradient directions.

[0007] In accordance with another aspect of the present invention a method and apparatus for identifying objects in an image are provided. In accordance with this aspect, a gradient magnitude is determined for each pixel in the image. A gradient direction for each pixel in the image is determined, the gradient direction being determined using a look up table.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Other objects and advantages of the invention will become apparent to those skilled in the art upon reading the following detailed description of preferred embodiments, in conjunction with the accompanying drawings, wherein like reference numerals have been used to designate like elements, and wherein:

[0009] FIG. 1 is a block diagram of the processing performed for identifying objects in an image in accordance with exemplary embodiments of the present invention.

[0010] FIG. 2 is a block diagram of the processing performed to produce a gradient magnitude image and a gradient direction image in accordance with exemplary embodiments of the present invention.

[0011] FIG. 3A illustrates gradient direction values in accordance with exemplary embodiments of the present invention.

[0012] FIG. 3B illustrates and exemplary convolution kernel for processing an image in accordance with exemplary embodiments of the present invention.

[0013] FIG. 4 is a table for encoding edge directions in accordance with exemplary embodiments of the present invention.

[0014] FIG. 5 is a block diagram of the processing performed on portions of the image in accordance with exemplary embodiments of the present invention.

[0015] FIG. 6 is a block diagram of the processing performed for a binary OR operation in accordance with exemplary embodiments of the present invention.

[0016] FIG. 7 is a block diagram of the processing performed for determining the number of different directions in a portion of an image in accordance with exemplary embodiments of the present invention.

[0017] FIG. 8 illustrates an exemplary curve representing an empirically derived confidence of an object determined as a function of the count of different gradient directions in accordance with the present invention.

[0018] FIGS. 9A and 9B illustrate gradient directions in accordance with conventional image processing techniques.

[0019] FIGS. 10A and 10B illustrate gradient directions in accordance with exemplary embodiments of the present invention.

[0020] FIG. 11 illustrates and exemplary look-up table in accordance with exemplary embodiments of the present invention.

DETAILED DESCRIPTION

[0021] In accordance with exemplary embodiments of the present invention, portions of an image are processed to determine the number of different gradient directions present in the portion of the image. Through empirical analysis, it has been determined that closed objects, or nearly closed objects, in an image typically includes a predetermined number of different edge directions. For example, if the directions in an edge direction image are quantitized to one of eight unique directions, an object will normally comprise six, seven or eight different edge directions. It should be recognized that a quantitization of eight edge directions is merely exemplary, and that the present invention is equally applicable to other quantitizations, e.g., 16 or 32 edge directions. If other quantitizations are employed, the number

2

of different edge directions used for identifying objects in an image can be determined by one of ordinary skill in the art through routine empirical analysis.

[0022] **FIG. 1** is a block diagram of the processing performed for identifying objects in an image in accordance with exemplary embodiments of the present invention. An input image **110** is provided to processing block **120**, where a gradient operator determines the gradient direction and gradient magnitude of each pixel in the input image **110**. After determining the gradient magnitude and gradient direction, a threshold function can be determined as a function of the magnitude and applied to the gradient magnitude to remove background noise (e.g., based on empirical information). If the gradient magnitude of a particular pixel is below the threshold, the gradient magnitude and gradient direction for that pixel are discarded. Conversely, if the gradient magnitude is above the threshold, the gradient magnitude and gradient direction are retained. The values of each pixel in the gradient direction image are encoded by processing block **130**. Processing block **140** performs a box sum OR operation on the encoded edge direction image. Specifically, processing block **140** operates on portions of the encoded edge direction image to determine whether the portion contains an object. The size of the portion which is processed is selected to equal the size of objects which it is desired to identify in the image.

[0023] For each portion of the image a horizontal binary OR operation **142** is performed, followed by a vertical binary OR operation **144**. The result of these operations are input to an edge count lookup table in processing block **146**, which outputs a value indicating the number of different edge directions present in the portion of the image processed by processing block **140**. Specifically, the output can include the thresholded gradient magnitude and gradient direction image with an indication of the number of different directions present in each portion of the image or an indication of which portions of the image contain objects. The output can be provided on a display or in printed form. If this processing is part of an automated system, the output can be in the form of coordinates of where objects are located in the images.

[0024] **FIG. 2** is a block diagram of the processing performed by processing block **120**. Specifically, an input image is processed by a gradient operator to output a gradient direction image **220** and a gradient magnitude image **230**. In accordance with exemplary embodiments of the present invention, a Sobel operator is employed as the gradient operator. However, one of ordinary skill in the art will recognize that there are many other types of gradient operators which may be employed. **FIG. 3A** illustrates the values corresponding to each gradient direction. **FIG. 3B** illustrates the convolution kernel employed for processing the image to produce the gradient direction image and the gradient magnitude image. Specifically, a gradient x vector is calculated in accordance with the following equation:

$$D=a+2*d+g-c-2*f-i \qquad (1)$$

[0025] A gradient y vector is calculated in accordance with the following equation:

$$D=a+2*b+c-g-2*h-i \qquad (2)$$

[0026] Using the gradient x and y vectors, the gradient magnitude and gradient direction are calculated as follows:

$$\text{Magnitude} = \sqrt{D_x^2 + D_y^2} \qquad (3)$$

$$\text{Direction} = \text{arcTan}\left(\frac{D_y}{D_x}\right) \qquad (4)$$

[0027] Returning now to **FIG. 1**, the gradient direction image is encoded. **FIG. 4** illustrates the encoding performed on the gradient direction image. Specifically, each edge direction is encoded into an eight bit binary value as illustrated in **FIG. 4**. As will be described in more detail below, a particular encoded binary value for a particular edge direction is selected such that when a plurality of gradient directions are logically ORed, it is possible to determine the number of different edge directions from the resultant binary value.

[0028] **FIG. 5** illustrates the processing performed by box sum OR processing block **140** in detail. A portion of the encoded gradient directions image **510** equal to the size of the desired object be identified is input to processing block **520** where a binary OR operation is performed in the vertical direction. This is followed by a binary OR operation in the horizontal direction in processing block **530** which outputs a binary box OR **540**. Since the edge directions are encoded into one of the eight bits of a byte, the individual edge directions are added together by these binary OR operations. There is a significant improvement by performing the binary OR operation first in the vertical direction and then in the horizontal direction. In alternative embodiments, the vertical OR operation can be performed before the horizontal OR operation. For example, if a portion of the image is 32 pixels by 32 pixels, this would normally require 1023 OR's operations. If they are separated in vertical and horizontal convolutions, this requires 62 OR operations, resulting in a saving of about 16 to 1 operations. As will be described below in connection with **FIG. 6**, using a binary OR-ing this operation count can be reduced further.

[0029] **FIG. 6** illustrates the processing performed by the binary OR-ing processing in accordance with exemplary embodiments of the present invention. As illustrated in **FIG. 6**, all encoded directions are ORed with their adjacent neighbors. This results in layer A2. Next, an OR operation is performed in layer A2 for a particular pixel with its three away neighbor, resulting in layer A3. The information in each of the pixels of layer A3 contains four OR'ed values thus saving one step. Next, in level A3, a value is OR'ed with its seven away neighbor, thus resulting in layer A4. The information in each of the pixels of layer A4 contains eight OR'ed values. This process will continue until the vertical or horizontal box size of the portion of the image is achieved. It will be noted that all the level OR operations are performed on every resulting pixel, and that there are the same number of pixels at each level. It will also be recognized that the complete sets of gates that represent this operation are not illustrated in **FIG. 6**, and if they were, there would be 64 gates not just the 15 illustrated. Performing the binary OR operation in this manner for a 32 pixel vertical portion of an image is from 32 OR operations down to 6 OR operation. Accordingly, operating on a 32 pixel by 32 pixel portion of

the image, the number of OR operations goes from 1023 to 12 OR's for a total reduction of 83 to 1 operations.

[0030]   FIG. 7 illustrates the binary OR-ing of two gradient directions in accordance with exemplary embodiments of the present invention. In the example illustrated in **FIG. 7**, a gradient direction of seven has been encoded into the binary value of 01000000, and a gradient direction of three has been encoded into a binary value of 00000100. These binary values are logically ORed to form the binary value of 01000100, equivalent to 136. Using a look up table, it is determined that the value 136 corresponds to two different gradient directions.

[0031]   Once the number of different gradient directions are determined, a confidence value of the likelihood that the portion of the image identified is generated as containing an object actually contains an object is generated. **FIG. 8** illustrates an exemplary curve representing an empirically derived confidence of an object determined as a function of the count of different gradient directions. As illustrated in **FIG. 8**, as the number of different gradient directions increase, the confidence value that the particular portion of the image is actually an object increases. It should be recognized that the use of the curve can be replaced by an appropriate classifier algorithm.

[0032]   As discussed above, the present invention employs the conventional Sobel operator to determine the gradient directions. However, the conventional Sobel operator described in accordance with equations 1 through 4 above, requires 11 additions, 6 multiplications, 1 division, 1 square root, and 1 inverse tangent. Conventionally, the number of operations are decreased by performing the Sobel operation in accordance with equations 5 through 7 as follows:

$$D_x=a+2*(d-f)+g-c-i \tag{5}$$
$$D_y=a+2*(b-h)+c-g+i \tag{6}$$
$$\text{Magnitude}=abs(D_x)+abs(D_y) \tag{7}$$

[0033]   As illustrated in **FIG. 5**, the number of multiplications are reduced from 6 to 4 by grouping the d–f term and the b–h term. In addition, as illustrated in equation 7, by employing the absolute value of the x vector and the y vector, the square root operation can be eliminated. The gradient direction will still be calculated in accordance with equation 4.

[0034]   It can be desirable to further reduce the number of operations required to determine the gradient direction. Prior to describing the exemplary technique for reducing the number of operations in accordance with the present invention, a review of the gradient directions of the conventional Sobel operation will be described in connection with **FIGS. 9A and 9B**. As illustrated in FIG. 9B, the x axis bisects the gradient directions **1** and **5**, and the y axis bisects the gradient directions **3** and **7**. Accordingly, for a particular x vector and a particular y vector, a gradient direction of one will include those values 22.5° above or below the x axis. As illustrated in **FIGS. 8A and 8B**, the 22.5° about the x axis also exists for gradient direction **5**, and about the y axis for gradient directions **3** and **7**.

[0035]   By rotating the boundaries of the gradient directions 22.5°, the calculations for the Sobel operation can be simplified. **FIGS. 10A and 10B** illustrate this rotation. Accordingly, as illustrated in **FIG. 10A**, gradient direction **1** now corresponds to the portion between the x axis and 45°

above the x axis and gradient direction **8** now corresponds to the area between x axis and negative 45° from the x axis. Using this 22.5° rotation, the x vector and the y vector can be calculated in accordance with the following equations:

$$D_x=l+k+[(d-f)<<1] \tag{8}$$
$$D_y=l-k+[(b-h)<<1] \tag{9}$$

wherein l=a+i and k=g−c, and the double < represents a one bit binary shift to the right.

[0036]   Using the x vector and the y vector, a lookup table in **FIG. 11** is used to determine the gradient direction. Specifically, if the y direction vector is greater than or equal to zero, the third bit of a three bit value is set equal to one, otherwise this bit is set equal to zero. If the x vector is greater than or equal to zero, the second bit of a three bit binary value is set equal to one, otherwise this bit is set equal to zero. Finally, if the absolute value of the x vector is greater than the absolute value of the y vector, then the first bit of a three bit binary value is set equal to one, otherwise it is set equal to zero. Accordingly, three bit binary words will be calculated for each gradient direction. These three words are binary OR'ed to result in a new 3 bit binary word. This new three bit binary word can be used to determine the gradient direction as illustrated in **FIG. 10**. Accordingly, by using the procedures described above for the Sobel operation, requires only 10 additions and two one bit shift rights, and zero multiplications, divisions, square roots, and inverse tangents. Accordingly, it can be seen that the 22.5° rotation can greatly increase the processing speed, and reduce the processing power, required to determine the gradient direction.

[0037]   Using the 22.5° rotation described above provides an adequate approximation of the gradient direction, this approximation can be improved. Specifically, using equations 10 through 15 below, takes advantage of the decrease in operations achieved by the 22.5° rotation, while compensating for this rotation.

$$D'_x=D_x*15137-D_y*6270 \tag{10}$$
$$D'_y=D_x*6270+D_y*15137 \tag{11}$$
$$d1=[CMP(D'_x, 0)]>>1 \tag{12}$$
$$d2=CMP(D'_y, 0) \tag{13}$$
$$d3=[CMP(abs(D'_y,abs(D'_y))]>>2 \tag{14}$$
$$\text{Direction}=LUT((d3)!(d2)!(d1)) \tag{15}$$

where CMP represents a comparison operation, LUT represents a lookup table operation, and an exclamation point represents a binary OR operation.

[0038]   In equations 10 and 11 the values 15,137 and 6,270 are employed to compensate for the 22.5° binary shift. Specifically, the value of 15,137 represents the cosine of 22.5° times a scaling factor, and the value 6,270 represents the sine of 22.5° times a scale factor.

[0039]   For ease of understanding, the present invention has been generally described as performing processing and logical operations. The processing and logical operations can be implemented using a variety of mechanisms including, but not limited to, Application Specific Integrated Circuits (ASICs), a microprocessor which executes software code, and hard-wired logic circuits. Moreover, the tables described herein can be stored in a variety of devices including buffers, caches, Random Access Memory (RAM), Read Only Memory (ROM), and the like.

[0040] The present invention has been described with reference to several exemplary embodiments. However, it will be readily apparent to those skilled in the art that it is possible to embody the invention in specific forms other than those of the exemplary embodiments described above. This may be done without departing from the spirit of the invention. These exemplary embodiments are merely illustrative and should not be considered restrictive in any way. The scope of the invention is given by the appended claims, rather than the preceding description, and all variations and equivalents which fall within the range of the claims are intended to be embraced therein.

1. A method for identifying objects in an image comprising:

   processing the image with a gradient operator to produce a gradient magnitude and direction for each pixel;

   determining a number of different gradient directions in a portion of the processed image; and

   identifying the portion of the processed image as an object if the number of different gradient directions exceeds a threshold number of gradient directions.

2. (canceled)

3. The method of claim 21, wherein each possible gradient direction is associated with a unique binary value, and wherein the gradient direction for each pixel is encoded into one of the unique binary values.

4. The method of claim 3, wherein a binary OR operation of any number of the unique binary values results in a binary value which uniquely identifies a number of different gradient directions employed in the binary OR operation.

5. The method of claim 1, wherein the image is a monochrome image with image intensities on various gray levels.

6. The method of claim 1, wherein the image is processed using a Sobel operator.

7. The method of claim 1, wherein the step of processing the image comprises:

   determining a gradient magnitude for each pixel in the image; and

   determining a gradient direction for each pixel in the image, the gradient direction being determined using a look up table.

8. A method for identifying objects in an image comprising:

   determining a gradient magnitude for each pixel in the image;

   determining a gradient direction for each pixel in the image, the gradient direction being determined using a look up table.

9. The method of claim 8, wherein the step of determining a gradient direction comprises:

   calculating a first and second gradient vector for each pixel;

   determining whether the first and second gradient vectors are positive integers; and

   determining whether an absolute value of the first gradient vector is greater than an absolute value of the second gradient vector.

10. The method of claim 9, further comprising:

   setting a first binary value of a first binary word to equal one if the first gradient vector is a positive integer;

   setting a second binary value of a second binary word equal to one if the second gradient vector is a positive integer; and

   setting a third binary value of a third binary word equal to one if the absolute value of the first gradient vector is greater than the absolute value of the second gradient vector.

11. The method of claim 10, further comprising:

   performing a logic OR operation on the first, second and third binary words;

   determining a gradient direction based on a result of the logical OR operation.

12. A computer readable-medium containing a program which executes the steps of:

   processing an image with a gradient operator to produce a gradient magnitude and direction for each pixel;

   determining a number of different gradient directions in a portion of the processed image; and

   identifying the portion of the processed image as an object if the number of different gradient directions exceeds a threshold number of gradient directions.

13. (canceled)

14. The computer-readable medium of claim 12, wherein each possible gradient direction is associated with a unique binary value, and wherein the gradient direction for each pixel is encoded into one of the unique binary values.

15. The computer-readable medium of claim 14, wherein a binary OR operation of any number of the unique binary values results in a binary value which uniquely identifies a number of different gradient directions employed in the binary OR operation.

16. The computer-readable medium of claim 12, wherein the image is a monochrome image with image intensities on various gray levels.

17. The computer-readable medium of claim 12, wherein the image is processed using a Sobel operator.

18. The computer-readable medium of claim 12, wherein the step of processing the image comprises:

   determining a gradient magnitude for each pixel in the image; and

   determining a gradient direction for each pixel in the image, the gradient direction being determined using a look up table.

19. A computer-readable medium containing a program which executes the steps of:

   determining a gradient magnitude for each pixel in an image; and

   determining a gradient direction for each pixel in the image, the gradient direction being determined using a look up table.

20. The computer-readable medium of claim 19, wherein the step of determining a gradient direction comprises:

   calculating a first and second gradient vector for each pixel;

determining whether the first and second gradient vectors are positive integers; and

determining whether an absolute value of the first gradient vector is greater than an absolute value of the second gradient vector.

21. A computer-readable medium of claim 20, wherein the computer program performs the steps of:

setting a first binary value of a first binary word to equal one if the first gradient vector is a positive integer;

setting a second binary value of a second binary word equal to one if the second gradient vector is a positive integer; and

setting a third binary value of a third binary word equal to one if the absolute value of the first gradient vector is greater than the absolute value of the second gradient vector.

22. The computer-readable medium of claim 21, wherein the computer program performs the steps of:

performing a logic OR operation on the first, second and third binary words; and

determining a gradient direction based on a result of the logical OR operation.

23. A system comprising:

a processor which processes the image with a gradient operator to produce a gradient magnitude and direction for each pixel, determines a number of different gra-

dient directions in a portion of the processed image, and identifies the portion of the processed image as an object if the number of different gradient directions exceeds a threshold number of gradient directions; and

means for outputting the identified portion of the processed image.

24. (canceled)

25. The system of claim 23, wherein each possible gradient direction is associated with a unique binary value, and wherein the gradient direction for each pixel is encoded into one of the unique binary values.

26. The system of claim 25, wherein a binary OR operation of any number of the unique binary values results in a binary value which uniquely identifies a number of different gradient directions employed in the binary OR operation.

27. The system of claim 23, wherein the image is a monochrome image with image intensities on various gray levels.

28. The system of claim 23, wherein the image is processed using a Sobel operator.

29. The system of claim 23, wherein the processor processes the image with the gradient operator by determining a gradient magnitude for each pixel in the image, and by determining a gradient direction for each pixel in the image, the gradient direction being determined using a look up table.

* * * * *